TESTING A SIMPLE SYMMETRIC HYPOTHESIS BY A FINITE-MEMORY DETERMINISTIC ALGORITHM

Calvin Marion Anderson

LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIF. 93940

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

TESTING A SIMPLE SYMMETRIC HYPOTHESIS
BY A FINITE-MEMORY DETERMINISTIC ALGORITHM

by

Calvin Marion Anderson

Thesis Advisor: B. O. Shubert

September 1973

Approved for public release; distribution unlimited.

Testing a Simple Symmetric Hypothesis
by a
Finite-Memory Deterministic Algorithm

by

Calvin Marion Anderson
Major, United States Army
B.S., South Dakota School of Mines and Technology, 1962

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 1973

LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIF. 93940

ABSTRACT

A class of finite-memory deterministic algorithms is introduced and investigated. Optimum algorithms are found for a small number of states (up to 21) and an asymptotic bound on error probability is obtained for a large number of states. The algorithms provide their own stopping rule.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENT

The invaluable advice and assistance of Professor Bruno Shubert is gratefully acknowledged.

I. INTRODUCTION

Computers seem to grow more complex and more sophisticated on almost a continuous basis. A logical future step in computer development would be to provide the computer with some decision making capability. This capability of necessity would be limited by the size of the computer core. Such a property would be of particular value if the computer were designed to operate without human assistance. For example, exploration of the nearer stars could most easily be accomplished by unmanned spacecraft. Yet the immense distances preclude human involvement in any decision process. If a computer with a decision making capability were to be used within the spacecraft, it would almost certainly be small and of very limited core size. Such an automaton would be required to make decisions with minimum probability of error and constrained by available memory. One form of a decision process could probably be adapted from the statistical test of hypothesis.

In testing a hypothesis, the statistician normally forms the likelihood ratio and reduces the ratio to a sufficient statistic which is to be less than or greater than some constant k. For a sample of size n; α_n (Probability of Type I error) and β_n (Probability of Type II error) will exponentially approach zero as n becomes large. To apply the procedure at time n means that sufficient memory must be available at time n to record the observations $X_1, X_2, X_3, \ldots, X_n$. In even the simplest cases this memory must grow indefinitely with time. Summarizing the data in a sufficient statistic does not insure that the memory requirement will be reduced. The sufficient statistic is data reducing only in that it maps the observations from R^n to R, but the cardinality of the memory may be

at least as great. For example, suppose that an experimenter wishes to estimate μ for a Normal random variable and uses $\bar{x}$ as a sufficient statistic. Further suppose that in 99 repeated trials the experimenter finds that $\Sigma x = 1.000$. Then $\bar{x} = 1.0/99 = 0.0101....$, and the memory requirement has become infinite. It would be tempting to round off the statistic to some finite dimension; however, Cover [1] has shown that α_n and β_n then do not tend to zero. Additionally, the rounded off statistic may not converge to the same distribution as the estimated parameter. If constrained by finite memory, some other model must be devised. One possible approach is to use only the last k observations. This idea has been investigated by Robbins [2].

Although not originally intended as a method for testing hypotheses, Robbins' model maximizes the long run expected number of "successes" given two alternative courses of action and finite memory. Suppose that an experimenter has two coins and that he wishes to maximize the number of heads thrown during a sequence of tosses. The minimum variance unbiased estimator of p is $\bar{x}$ and as the number of repeated trials increases, the variance of the estimator goes to zero. Therefore if an experimenter had prior knowledge of the probability of heads for each coin he would use the coin with the greater probability of heads exclusively and know with certainty that

$$\text{Limit}_{n\to\infty} \frac{\text{number of heads in first n tosses}}{n} = \max(p_1, p_2)$$

where p_i = probability of heads for the i^{th} coin. Without prior knowledge, and constrained by a finite memory, the experimenter must decide which coin to use on the basis of the results of the previous r trials. Robbins formulates a decision rule in the proof of the following theorem:

"Define the rule R_r as follows: start tossing with coin 1. Stop

if the first toss is tail, otherwise continue tossing until the first run of r successive tails occurs and then stop. This defines the first block of tosses with coin 1. Now start tossing with coin 2 and apply the same rule, obtaining the first block of tosses with coin 2. Then start again with coin 1 and apply the same rule, obtaining the second block of tosses with coin 1, and so on indefinitely, thus generating an infinite sequence of tosses consisting of alternate blocks of tosses with coins 1 and 2. With rule R_r so defined, we assert that

$$\operatorname*{Limit}_{n\to\infty} \frac{\text{number of heads in first n tosses}}{n} = \frac{p_1 q_2^r + p_2 q_1^r}{q_1^r + q_2^r} ."$$

Note that

$$\operatorname*{Limit}_{r\to\infty} \frac{p_1 q_2^r + p_2 q_1^r}{q_1^r + q_2^r} = \max(p_1, p_2)$$

Using methods similar to the Robbins model, Cover [1] developed a 4-state memory algorithm for testing the hypothesis $p > p_0$ vs $p < p_0$, given a sequence of iid Bernoulli random variables. In the Cover model, the pair (T,Q) can take values in $\{0,1\}$. T keeps track of the currently favored hypothesis and Q records the results of the current run test. Two sequences $\{s_i\}_1^\infty$ and $\{r_i\}_1^\infty$ of positive intergers are considered. The sequences of observations are divided into blocks S_1, R_1, S_2, R_2,.... where S_1 denotes the first s_i observations, the next r_i by R_1, and so on. T is initially arbitrary; if all observations in a block S_i are equal to 1, Q is set to 1. If all observations in a block R_i are equal to 0, Q is set to 1. At the end of each block the currently favored hypothesis

is updated by the rule

$$T_n = 1 \quad \text{if } Q = 1 \text{ and } n \text{ is at the end of an S block}$$
$$= 0 \quad \text{if } Q = 1 \text{ and } n \text{ is at the end of an R block}$$
$$= T_{n-1} \quad \text{otherwise}$$

The lengths of the blocks of S's and R's are determined as a function of p_o, and Cover shows that the limiting probability of error under either hypothesis is zero. (See Figure 1).

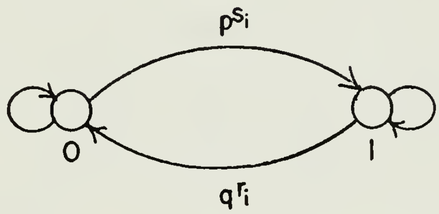


Figure 1.

A two state Markov chain where T can take on values in $\{0,1\}$.

Although the memory size in Cover's model is now finite the updating rule still depends on n.

The first genuine finite memory model has been proposed by Hellman and Cover [3],[4]. They proposed a family of algorithms of the type

$$T_n = f(T_{n-1}, x_n) \quad ; \quad f: \ \{1,2,\ldots,m\} \times \mathcal{X} \to \{1,2,\ldots,m\}$$
$$d_n = d(T_n) \quad ; \quad d: \ \{1,2,\ldots,m\} \to \{\mathcal{H},\mathcal{J}\}$$

T_n denotes the statistic at time n, x_n is the value of the n^{th} sample, f is a transition function and d is a decision function. Note that T is of finite memory since $T \in \{1,2,\ldots,m\}$; and given an initial value of the statistic, the sequence T_n forms a Markov chain over the state space $M = \{1,2,\ldots,m\}$. The goal is to minimize the expected asymptotic proportion of errors

$$P(e) = E\left\{\underset{n\to\infty}{\text{Limit}} \frac{1}{n}\sum_{i=1}^{n} e_i\right\}$$

where

$$e_i = \begin{cases} 1 & \text{if } d_i \neq H_{true} \\ 0 & \text{if } d_i = H_{true} \end{cases}$$

Hellman and Cover have established a lower bound for the proportion of of error. Let $\pi_{\mathcal{H}}$ and $\pi_{\mathcal{J}}$ denote the prior probabilities of the null and alternate hypotheses. Let $f_{\mathcal{H}}$ and $f_{\mathcal{J}}$ be the probability densities of the sample under the respective hypothesis with respect to a dominating measure. Define the likelihood ratio to be $\ell(x) \equiv f_{\mathcal{H}}(x)/f_{\mathcal{J}}(x)$. Let $\overline{\ell}$ denote the ess sup of the likelihood ratio and $\underline{\ell}$ the ess inf where the supremum and infimum are taken over all measureable sets with positive dominating measures. Define $\gamma \equiv \overline{\ell}/\underline{\ell}$. Then for an irreducible m-state automaton, $P(e) \geq P^*$ where

$$P^* = \frac{2(\pi_{\mathcal{H}}\pi_{\mathcal{J}}\gamma^{m-1})^{\frac{1}{2}} - 1,}{\gamma^{m-1} - 1} \qquad \text{if } \gamma^{m-1} \geq \max\left\{\pi_{\mathcal{H}}/\pi_{\mathcal{J}}, \pi_{\mathcal{J}}/\pi_{\mathcal{H}}\right\}$$

$$= \min\left\{\pi_{\mathcal{H}}, \pi_{\mathcal{J}}\right\} \qquad \text{otherwise.}$$

Hellman and Cover further prove that a reducible** (m+1)-state automaton obeys the same bound on P(e) as an irreducible m-state automaton.

If the prior probabilities of the null and alternate hypotheses are equally likely, that is if $\pi_{\mathcal{H}} = \pi_{\mathcal{J}} = \frac{1}{2}$, then for an irreducible m-state automaton

$$P^* = \frac{2(\frac{1}{2}\frac{1}{2}\gamma^{m-1})^{\frac{1}{2}} - 1}{\gamma^{m-1} - 1}$$

$$= \frac{\gamma^{\frac{1}{2}(m-1)} - 1}{\gamma^{(m-1)} - 1}$$

$$= \frac{1}{\gamma^{\frac{1}{2}(m-1)} + 1}$$

** We call the automaton reducible (irreducible) if the Markov chain $\{T_n\}$ is reducible (irreducible).

If the m-state automaton is reducible, the bound becomes at least as great as

$$P(e) \geq \frac{1}{\gamma^{\frac{1}{2}(m-2)} + 1}$$

In the case of the Bernoulli trials, consider the two hypotheses case

$$\mathcal{H} : p = p_{\mathcal{H}}$$

$$\mathcal{J} : p = p_{\mathcal{J}} \qquad \text{where } \pi_{\mathcal{H}} = \pi_{\mathcal{J}} = \tfrac{1}{2}$$

Without loss of generality it may be assumed that $p_{\mathcal{H}} > p_{\mathcal{J}}$ in which case $\overline{\ell} = p_{\mathcal{H}}/p_{\mathcal{J}}$; $\underline{\ell} = q_{\mathcal{H}}/q_{\mathcal{J}}$; and $\gamma = \overline{\ell}/\underline{\ell} = \dfrac{p_{\mathcal{H}} q_{\mathcal{J}}}{p_{\mathcal{J}} q_{\mathcal{H}}}$.

Further, if the hypothesis if symmetric, that is, if $p_{\mathcal{H}} = 1-p_{\mathcal{J}}$, then $\gamma = \left(\dfrac{p_{\mathcal{H}}}{q_{\mathcal{H}}}\right)^2$; for an irreducible m-state automaton

$$P(e) \geq \frac{1}{1 + \left(\frac{p_{\mathcal{H}}}{q_{\mathcal{H}}}\right)^{m-1}} \quad .$$

and for a reducible automaton

$$P(e) \geq \frac{1}{1 + \left(\frac{p_{\mathcal{H}}}{q_{\mathcal{H}}}\right)^{m-2}} \quad .$$

While the lower bound cannot be achieved except in degenerate cases, Hellman and Cover demonstrate an ϵ-optimal class of automata, that is, for every $\epsilon>0$ there exists an automaton such that $P(e) \leq P^* + \epsilon$.

Define:

$$\mathcal{H}_\epsilon = \{x \in \mathfrak{X} : \ell(x) \geq [(1/\overline{\ell}) + \epsilon]^{-1}\}$$

$$\mathcal{J}_\epsilon = \{x \in \mathfrak{X} : \ell(x) \leq (\underline{\ell} + \epsilon)\}$$

$$\mathcal{L}_\epsilon = \{x \in \mathfrak{X} : \ x \notin (\mathcal{H}_\epsilon \cup \mathcal{J}_\epsilon)\}$$

Let the transition function f, be specified as follows (see figure 2):

$$f(i,x) = \begin{cases} i+1 & x \in \mathcal{H}_\epsilon \\ i & x \in \mathcal{S}_\epsilon \\ i-1 & x \in \mathcal{T}_\epsilon \end{cases} \qquad \text{for } 2 \le i \le m-1$$

$$f(1,x) = \begin{cases} 2 & \text{with probability } \delta > 0 \text{ if } x \in \mathcal{H} \\ 1 & \text{otherwise} \end{cases}$$

$$f(m,x) = \begin{cases} m-1 & \text{with probability } k\delta > 0 \text{ if } x \in \\ m & \text{otherwise} \end{cases}$$

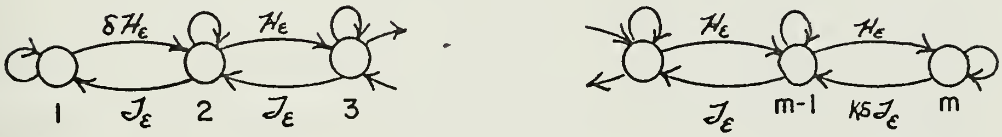


Figure 2.

Transitions are made to adjacent states only when the events $\mathcal{H}_\epsilon$ or $\mathcal{T}_\epsilon$ are observed. Thus the automaton enters an end state only on strong evidence to support that hypothesis. If δ is allowed to become arbitrarily small, then the automaton tends to leave the end state with a very low probability. Decisions made in the end states have the least probability of error, so as $\delta \to 0$, the P(e) should asymptotically approach P*.

While the Hellman-Cover algorithm is useful in producing sequences of decisions, the algorithm is not easily adapted to situations in which only a single decision is required. The irreducible automaton will asymptotically approach the lower bound for probability of error after a "large enough" number of observations; however, there is no easily defined rule which would specify when this number had been reached. It should also be noted that the Hellman-Cover automaton requires artificial randomization for transitions out of the end states. Some ancillary mechanism must be provided to achieve this desired randomization. In the case of a small computer, additional core storage would probably be

required. The closer P(e) is to approach P*, the smaller is the probability δ which must be generated - which requires even more additional core storage. It is therefore believed that there are strong pragmatic reasons for adopting an algorithm with absorbing states despite higher asymptotic probability of error.

In this paper a special class, a_n, of symmetric (2n + 3)-state algorithms with two absorbing states will be developed. Derivations and proofs within the paper are restricted to symmetric Bernoulli random variables, but it would also be feasible to extend these concepts to non-symmetric hypotheses and to distributions other than Bernoulli.

II. DESCRIPTION OF THE ALGORITHM

Let X_1, X_2, X_3,... denote a sequence of independent identically distributed Bernoulli random variables which can take on values H or T. Consider two hypotheses, $\mathcal{H}$ and $\mathcal{T}$ with equal prior probabilities and such that

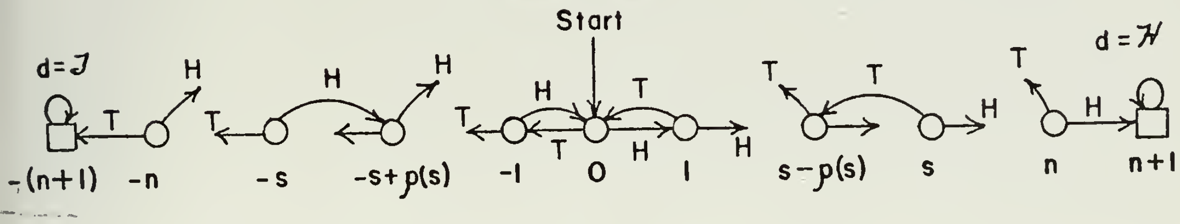
$$P(X_1 = H|\mathcal{H}) = P(X_1 = T|\mathcal{T}) = p, \text{ where } \tfrac{1}{2} < p < 1.$$

As the notation suggests, the sequence of random variables can be thought of as successive tosses of a coin which is biased towards Heads under hypothesis $\mathcal{H}$ or biased towards Tails under hypothesis $\mathcal{T}$.

Define the algorithm (M,f,d) (See Figure 3) such that $M = \{-(n+1), -n, \ldots, -1,0,1,\ldots n,n+1\}$ with $\pm(n+1)$ the two absorbing states and 0 the initial state; $d(n+1) = \mathcal{H}$, $d(-(n+1)) = \mathcal{T}$, otherwise arbitrary; and the transition function, f, such that

$$\begin{array}{lll}
f(s,H) = s+1, & f(s,T) = s-\rho(s) & \text{if } s = 1,2,\ldots,n \\
f(s,T) = s-1, & f(s,H) = s+\rho(s) & \text{if } s = -1,\ldots,-n \\
f(s,H) = 1, & f(s,T) = -1 & \text{if } s = 0 \\
f(s,H) = s, & f(s,T) = s & \text{if } s = \pm(n+1)
\end{array}$$

The integers $\rho(1),\ldots,\rho(n)$ satisfy the inequality $1 \leq \rho(s) \leq s$.



An Algorithm $f = \langle\rho(1),\ldots,\rho(n)\rangle \in a_n$

Figure 3.

The specific form of the algorithm (M,f,d) will henceforth be denoted $f = \langle \rho(1),\ldots,\rho(n)\rangle$. Figure 4 shows the algorithm and the transition matrix for the case when $n = 4$ and $f = \langle 1,1,2,2\rangle$.

	5	4	3	2	1	0	-1	-2	-3	-4	-5
5	1	0	0	0	0	0	0	0	0	0	0
4	p	0	0	q	0	0	0	0	0	0	0
3	0	p	0	0	q	0	0	0	0	0	0
2	0	0	p	0	q	0	0	0	0	0	0
1	0	0	0	p	0	q	0	0	0	0	0
0	0	0	0	0	p	0	q	0	0	0	0
-1	0	0	0	0	0	p	0	q	0	0	0
-2	0	0	0	0	0	0	p	0	q	0	0
-3	0	0	0	0	0	0	p	0	0	q	0
-4	0	0	0	0	0	0	0	p	0	0	q
-5	0	0	0	0	0	0	0	0	0	0	1

Figure 4: The transition matrix for the case where $n = 4$ and $f = \langle 1,1,2,2\rangle$.

III. THE ϵ-OPTIMAL STOCHASTIC AUTOMATON

Although the class a_n consists of deterministic algorithms only, it can be easily shown that with randomization, the Hellman-Cover lower bound could be approached arbitrarily closely. With the algorithm (M,f,d), the probability of error can be written

$$P_e = \tfrac{1}{2}\,\Pr(\text{absorption at } -(n+1)\,|\,\mathcal{H}) + \tfrac{1}{2}\,\Pr(\text{absorption at } n+1\,|\,\mathcal{T}).$$

Let $P_i^j(\mathcal{H})$ denote the absorption in state i without return to j given that hypothesis $\mathcal{H}$ is true and $P_i^j(\mathcal{T})$ denote the corresponding probability given $\mathcal{T}$ where $i = \pm(n+1)$. Then since 0 is a recurrent event

$$P_e = \tfrac{1}{2}\,P^o_{-(n+1)}(\mathcal{H}) + \tfrac{1}{2}\,P^o_{n+1}(\mathcal{T}) \quad \text{which by symmetry}$$

$$= \quad P^o_{-(n+1)}(\mathcal{H})$$

Given $\mathcal{H}$, we know that $P^o_{-(n+1)}(\mathcal{H}) + P^o_{n+1}(\mathcal{H}) = 1$, so

$$P_e = \frac{P^o_{-(n+1)}(\mathcal{H})}{P^o_{-(n+1)}(\mathcal{H}) + P^o_{n+1}(\mathcal{H})}$$

$$= \left[1 + \frac{P^o_{n+1}(\mathcal{H})}{P^o_{-(n+1)}(\mathcal{H})}\right]^{-1} \tag{1}$$

Let $f = \langle 1,2,3,\ldots,n\rangle$ and $1 > \delta > 0$ and consider the absorption at n+1. If a Head is observed, move to the next higher state with probability δ, otherwise remain in that state. If a Tail if observed, return to 0 since $\rho(s) = s$ for all s. Since return to 0 is a recurrent event,

$$P^o_{n+1}(\mathcal{H}) = p^{n+1}\delta^{n+1} + \binom{n+1}{1}p^{n+2}\delta^{n+1}(1-\delta) + \binom{n+2}{2}p^{n+3}\delta^{n+1}(1-\delta)^2 + \ldots$$

$$= p^{n+1}\delta^{n+1}\left[1 + \binom{n+1}{1}p(1-\delta) + \binom{n+2}{2}p^2(1-\delta)^2 + \ldots\right.$$

$$= p^{n+1}\delta^{n+1} \sum_{k=0}^{\infty} \binom{n+k}{k} p^k (1-\delta)^k$$

$$= p^{n+1}\delta^{n+1} [1 - p(1-\delta)]^{-n}$$

$$= p^{n+1}\delta^{n+1} (q + \delta p)^{-n}$$

Similarly,

$$P^o_{-(n+1)}(\mathcal{H}) = q^{n+1}\delta^{n+1} (p + \delta q)^{-n}$$

Thus

$$P_e = \left[1 + \frac{p^{n+1}(p + \delta q)^n}{q^{n+1}(q + \delta p)^n}\right]^{-1}$$

Taking the limit as $\delta \to 0$, we have

$$P_e = \left[1 + \left(\frac{p}{q}\right)^{2n+1}\right]^{-1}$$

which is the Hellman-Cover lower bound. Unfortunately this ϵ-optimal automaton is of little practical use since the expected time to absorption becomes infinite.

IV. DETERMINATION OF ERROR PROBABILITY

To obtain an explicit expression for the probability of error in terms of the algorithm, we will first prove the following proposition.

Proposition 1: Let $f = \langle \rho(1), \ldots, \rho(n) \rangle$

Then $$P_e = \left[1 + \left(\frac{p}{q}\right)^{n+1} R_n(f) \right]^{-1}, \quad R_n(f) = \frac{F_n(q,f)}{F_n(p,f)} \tag{2}$$

Where $F(x,f)$ is a polynomial in x of degree less than or equal to n and with integral coefficients. With initial conditions $F_{-1} = 0$ and $F_o = 1$, these polynomials satisfy the recurrence relationship

$$F_n(x,f) = F_{n-1}(x,f) - (1-x)x^{\rho(n)} F_{n-1-\rho(n)}(x,f) \tag{3}$$

where $n = 1,2,\ldots$.

Proof:

From (1) we know that

$$P_e = \left[1 + \frac{P^o_{n+1} \quad (\mathcal{H})}{P^o_{-(n+1)} (\mathcal{H})} \right]^{-1}$$

Next note that

$$P^o_{n+1} \ (\mathcal{H}) = pP'_{n+1}(\mathcal{H}) \quad \text{and that}$$

$P^1_{n+1} \ (\mathcal{H}) = p\nu_{1,n}$ where $\nu_{1,n}$ is the expected number of visits to n before a visit to n+1 or 0 given that the chain was started in state 1. From basic properties of Markov chains we know that $\nu_{1,n}$ [5] is the $(1,n)^{th}$ entry of the fundamental matrix $M = (I-Q)^{-1}$ where $Q = [p_{ij}]$ is an n X n matrix with entries $p_{ij} = p$ if $j = i+1$, $p_{ij} = q$ if $j = i-\rho(i)$; and $p_{ij} = 0$ otherwise. (See Figure 5).

$$\begin{bmatrix} 0 & p & 0 & 0 & 0 & 0 \\ q & 0 & p & 0 & 0 & 0 \\ 0 & q & 0 & p & 0 & 0 \\ 0 & q & 0 & 0 & p & 0 \\ 0 & q & 0 & 0 & 0 & p \\ 0 & 0 & q & 0 & 0 & 0 \end{bmatrix}$$

Figure 5: The matrix Q for the case when n = 6 and f = $\langle 1,1,1,2,3,3 \rangle$.

Applying the formula for the inverse of a matrix, we have

$$\nu_{1,n} = (-1)^{n+1} \left| (I-Q)_{(n,1)} \right| \, |I-Q|^{-1}$$

where $| \cdot |$ is the determinant operation and $|(I-Q)_{(n,1)}|$ is the determinant of the $(1,n)^{th}$ cofactor transposed. By deleting the first column and n^{th} row of the matrix (I-Q), the submatrix $(I-Q)_{(n,1)}$ is lower triangular with entries -p on the diagonal. Hence its determinant is equal to $(-p)^{n-1}$. Substituting,

$$\begin{aligned} \nu_{1,n} &= (-p)^{n-1}(-1)^{n+1} |I-Q|^{-1} \\ &= p^{n-1} |I-Q|^{-1} \end{aligned}$$

If we denote $|I-Q| = F_n(p,f)$ and repeat the entire argument with p and q interchanged, then

$$\nu_{1,n} = p^{n-1} [F_n(p,f)]^{-1}$$

$$\nu_{-1,-n} = q^{n-1} [F_n(q,f)]^{-1}$$

Multiplying and substituting into (1), we have that

$$P_e = \left[1 + \left(\frac{p}{q}\right)^{n+1} \frac{F_n(q,f)}{F_n(p,f)} \right]^{-1}$$ which was to be proved.

To establish the recurrance formula for F_n (see Figure 6), expand the determinant along the n^{th} row. If $\rho(n) = n$, then the n^{th} row has all zeroes except a 1 on the diagonal; $F_n = F_{n-1}$ and (3) holds. If $\rho(n) < n$, expansion along the n^{th} column gives

$$F_n(p,f) = F_{n-1}(p,f) + p|I-Q|_{(n-1,n)}$$

Expand $|I-Q|_{(n-1,n)}$ along the last column and repeating this $\rho(n)-1$ times yields

$$F_n(p,f) = F_{n-1}(p,f) + p^{\rho(n)} D$$

The determinant D has all zeroes in the last row except the diagonal entry which is -q. Therefore,

$$D = -qF_{n-1-\rho(n)}(p,f) \quad \text{and}$$

$$F_n(p,f) = F_{n-1}(p,f) - qp^{\rho(n)}F_{n-1-\rho(n)}(p,f)$$

Checking the initial conditions completes the proof.

$$F_5(p,f) = \begin{vmatrix} 1 & -p & 0 & 0 & 0 \\ -q & 1 & -p & 0 & 0 \\ -q & 0 & 1 & -p & 0 \\ 0 & -q & 0 & 1 & -p \\ 0 & -q & 0 & 0 & 1 \end{vmatrix}$$

$$F_5(p,f) = F_4(p,f) + p\begin{vmatrix} 1 & -p & 0 & 0 \\ -q & 1 & -p & 0 \\ -q & 0 & 1 & -p \\ 0 & -q & 0 & 0 \end{vmatrix}$$

$$= F_4(p,f) + p^2\begin{vmatrix} 1 & -p & 0 \\ -q & 1 & -p \\ 0 & -q & 0 \end{vmatrix}$$

$$= F_4(p,f) + p^3\begin{vmatrix} 1 & -p \\ 0 & -q \end{vmatrix}$$

$$= F_4(p,f) + p^3 qF_1(p,f)$$

Figure 6. The recurrance relationship, $f = \langle 1,1,2,2,3 \rangle$.

Next, call

$$R_n^* = \max_{f \in a_n} R_n(f) = R_n(f^*)$$

Then from (2)

$$P_e^* = \min P_e = \left[1 + \left(\frac{p}{q}\right)^{n+1} R_n^*\right]^{-1}$$

If we take f = 1,2,3,...,n , then $R_n(f) = 1$ so that $R_n^* \geq 1$.

The Hellman-Cover lower bound states that

$$\left[1 + \left(\frac{p}{q}\right)^{n+1} R_n^*\right]^{-1} \geq \left[1 + \left(\frac{p}{q}\right)^{2n+1}\right]^{-1}$$

which implies that $\left(\frac{p}{q}\right)^{n+1} R_n^* \leq \left(\frac{p}{q}\right)^{2n+1}$ and $R_n^* \leq \left(\frac{p}{q}\right)^{n}$

Thus $1 \leq R_n^* \leq \left(\frac{p}{q}\right)^{n}$.

V. REFINING THE BOUNDS FOR R_n

In the preceeding paragraph, it was shown that $1 \le R_n{*} \le \left(\frac{p}{q}\right)^n$. It would now be useful to see if $R_n{*}$ exists in a form which can be expressed as a limit for large n. To do so, proposition 2 is proved.

Proposition 2:

Let $f_i = \langle \rho_i(1),\ldots,\rho_i(n_i)\rangle \epsilon\, a_{n_i}$ for $i = 1,2$, and $f = \langle \rho(1),\ldots,\rho(n_1 + n_2)\rangle \epsilon\, a_{n_1+n_2}$ be such that $\rho_2(s) < s$ for $s > 1$. Then $R^*_{n_1+n_2} \ge R_{n_1}(f_1)R_{n_2}(f_2)$.

Proof:

$$\text{Let } \rho(k) = \begin{cases} \rho_1(k) & \text{if } k = 1,\ldots,n_1 \\ n_1 + 1 & \text{if } k = n_1 + 1 \\ \rho_2(k-n_1) & \text{if } k = n_1 + 2,\ldots,n_1 + n_2 \end{cases}$$

First prove by induction on n_2 that

$$F_{n_1+n_2}(x,f) = F_{n_1}(x,f_1)F_{n_2}(x,f_2) \tag{4}$$

If $n_2 = 1$, then $F_{n_2} = 1$. But since $\rho(n_1+1) = n_1+1$, we know that (4) holds. If (4) is true for all $n_2 \le n$, then if $n_2 = n+1$ from the recurrance relationship we know that

$$F_{n_1+n_2}(x,f) = F_{n_1+n}(x,f) - x^{\rho_2(n_2)}(1-x)F_{n_1+n-\rho_2(n_2)}(x,f)$$

By the induction hypothesis,

$$F_{n_1+n}(x,f) = F_{n_1}(x,f_1)F_n(x,f_2) \quad \text{and}$$

$$F_{n_1+n-\rho_2(n_2)}(x,f) = F_{n_1}(x,f_1)F_{n-\rho_2(n_2)}(x,f_2)$$

If $\rho_2(n_2) \le n$, then

$$F_{n_1+n_2}(x,f) = F_{n_1+n}(x,f) - x^{\rho_2(n_2)}(1-x)F_{n_1+n-\rho_2(n_2)}(x,f)$$

$$= F_{n_1}(x,f_1)F_n(x,f_2) - x^{\rho_2(n_2)}(1-x)F_{n_1}(x,f_1)F_{n-\rho_2(n_2)}(x,f_2)$$

$$= F_{n_1}(x,f_1)[F_n(x,f_2) - x^{\rho_2(n_2)}(1-x)F_{n-\rho_2(n_2)}(x,f_2)]$$

$$= F_{n_1}(x,f_1)F_{n_2}(x,f_2)$$

which completes the proof of (4).

Equation (4) has been proved true which implies that

$R^*_{n_1+n_2} \geq R_{n_1+n_2}(f) = R_{n_1}(f_1)R_{n_2}(f_2)$ which completes the proof.

Next note that once an algorithm has been found such that $R(f) > 1$ and $\rho(s) < s$ for $s > 1$, that $\ln R^*_{n_1+n_2} \geq \ln R_{n_1}(f_1) + \ln R_{n_2}(f_2)$. If it is assumed that all optimal algorithms have this property from some n on, then $\ln R_n^*$ will be a monotonically increasing convex function. From the Hellman-Cover bound,

$$R_n^* \leq \left(\frac{p}{q}\right)^n$$

$$\ln R_n^* \leq n \ln\left(\frac{p}{q}\right)$$

$$\frac{\ln R_n^*}{n} \leq \ln\left(\frac{p}{q}\right)$$

If $\ln R_n^*$ is monotonically increasing, then the $\lim_{n\to\infty} \frac{\ln R_n^*}{n}$ exists, is positive, and is bounded above by $\ln \frac{p}{q}$.

To improve of the lower asymptotic bound for R_n^*, consider the algorithm $f = \langle \rho(1),\ldots,\rho(n)\rangle \in a_n$, with $\rho(s) = 1$ for $s = 1,\ldots,k$ and $\rho(s) = s-k$ for $s = k+1,\ldots,n$; where $1 \leq k \leq n$. From the recurrance relationship for F, we have

$$F_n(x) = F_{n-1}(x) - (1-x)x^{n-k}F_{k-1}(x)$$

$$F_{n-1}(x) = F_{n-2}(x) - (1-x)x^{n-k-1}F_{k-1}(x)$$

$$\vdots$$

$$F_{k+1}(x) = F_k(x) - (1-x)xF_{k-1}(x)$$

By substitution,

$$F_n(x) = F_k(x) - (1-x)(x+x^2+ \; . \; . \; . \; +x^{n-k}) \; F_{k-1}(x) \tag{5}$$

To establish an expression for $F_k(x)$, we will prove by induction that

$$F_j(x) = \left[x^{j+1} - (1-x)^{j+1}\right](2x-1)^{-1} \quad \text{for } j = 1,\ldots,k.$$

Proof:

Note that $F_1(x) = \dfrac{x^2-(1-x)^2}{2x-1} = \dfrac{2x-1}{2x-1} = 1.$

$$F_2(x) = \frac{x^3-(1-x)^3}{2x-1} = \frac{2x^3-3x^2+3x-1}{2x-1} = 1-x+x^2.$$

By the induction hypothesis,

$$F_j(x) = \left[x^{j+1}-(1-x)^{j+1}\right] (2x-1)^{-1} \qquad \text{for } j < k$$

$$F_{j-1}(x) = \left[x^j - (1-x)^j\right] (2x-1)^{-1}$$

From the recurrance relationship,

$$\begin{aligned}
F_{j+1}(x) &= F_j(x) - x(1-x)F_{j-1}(x) \\
&= \frac{x^{j+1}-(1-x)^{j+1}-x(1-x)[x^j-(1-x)^j]}{2x-1} \\
&= \frac{x^{j+1}-(1-x)^{j+1}-x^{j+1}+x^{j+2}+x(1-x)^{j+1}}{2x-1} \\
&= \frac{x^{j+2}-(1-x)^{j+2}}{2x-1}
\end{aligned}$$

which completes the proof of (6).

From (5) and (6)

$$\begin{aligned}
F_n(q,f) &= \frac{q^{k+1}-p^{k+1}-(1-q)(q+q^2+ \; . \; . \; . \; +q^{n-k})(q^k-p^k)}{q-p} \\
&= \frac{q^{k+1}-p^{k+1}+(q^{n-k+1}-q)(q^k-p^k)}{q-p} \\
&= \frac{p^k(q-p) + q^{n-k+1}(q^k-p^k)}{q-p} \\
&= p^k + q^{n-k+1}(p^k-q^k)(p - q)^{-1}
\end{aligned}$$

Similarly,

$$F_n(p,f) = q^k + p^{n-k+1}(p^k - q^k)(p - q)^{-1} \quad \text{and}$$

$$R_n(f) = \frac{p^k + q^{n-k+1}(p^k - q^k)(p - q)^{-1}}{q^k + p^{n-k+1}(p^k - q^k)(p - q)^{-1}}.$$

If $\alpha = \frac{k}{n}$ remains constant, then

$$R_n(f) = \frac{p^{\alpha n} + q^{n-\alpha n+1}(p^{\alpha n} - q^{\alpha n})(p - q)^{-1}}{q^{\alpha n} + p^{n-\alpha n+1}(p^{\alpha n} - q^{\alpha n})(p - q)^{-1}}$$

$$= \left(\frac{p}{q}\right)^{\alpha n} \frac{1 + p^{-\alpha n}q^{n(1-\alpha)}(p^{\alpha n} - q^{\alpha n})\frac{q}{p-q}}{1 + q^{-\alpha n}p^{n(1-\alpha)}(p^{\alpha n} - q^{\alpha n})\frac{p}{p-q}}$$

$$= \left(\frac{p}{q}\right)^{\alpha n} \frac{1 + q^{n}(q^{-\alpha n} - p^{-\alpha n})\frac{q}{p-q}}{1 + p^{n}(q^{-\alpha n} - p^{-\alpha n})\frac{p}{p-q}}$$

$$= \left(\frac{p}{q}\right)^{\alpha n} \frac{1 + q^{n}\frac{p^{\alpha n} - q^{\alpha n}}{p^{\alpha n}q^{\alpha n}}\frac{q}{p-q}}{1 + p^{n}\frac{p^{\alpha n} - q^{\alpha n}}{p^{\alpha n}q^{\alpha n}}\frac{p}{p-q}}$$

$$= \left(\frac{p}{q}\right)^{\alpha n} \frac{1 + \left(\frac{q^{1-\alpha}}{p^{\alpha}}\right)^{n}(p^{\alpha n} - q^{\alpha n})\frac{q}{p-q}}{1 + \left(\frac{p^{1-\alpha}}{q^{\alpha}}\right)^{n}(p^{\alpha n} - q^{\alpha n})\frac{p}{p-q}}$$

Let $U_n = \left(\frac{q^{1-\alpha}}{p^{\alpha}}\right)^{n}(p^{\alpha n} - q^{\alpha n}) = q^{n(1-\alpha)} - \left(\frac{q}{p^{\alpha}}\right)^{n}$

But $\frac{q}{p^{\alpha}} < 1$ for all α, so $\lim_{n\to\infty} U_n = 0$.

Let $V_n = \left(\frac{p^{1-\alpha}}{q^{\alpha}}\right)^{n}(p^{\alpha n} - q^{\alpha n})$

$$V_n = \left(\frac{p}{q^{\alpha}}\right)^{n} - p^{n(1-\alpha)}$$

$p^{n(1-\alpha)} \to 0$ for all α, but $\frac{p}{q^\alpha}$ may be greater than or equal to 1 depending on the choice of α.

If $\frac{p}{q^\alpha} = 1$,

$$\ln \frac{p}{q^\alpha} = 0; \quad \ln p - \alpha \ln q = 0; \; \alpha = \frac{\ln p}{\ln q}$$

If $\frac{p}{q^\alpha} > 1$,

$$\ln p - \alpha \ln q > 0; \; \alpha > \frac{\ln p}{\ln q}$$

If $\frac{p}{q^\alpha} < 1$,

$$\ln p - \alpha \ln q < 0; \; \alpha < \frac{\ln p}{\ln q}$$

Thus $\underset{n\to\infty}{\text{Limit}} \; V_n = \begin{Bmatrix} 0 \\ 1 \\ \infty \end{Bmatrix}$ if $\alpha \begin{Bmatrix} < \\ = \\ > \end{Bmatrix} \frac{\ln p}{\ln q}$

Substituting,

$$\underset{n\to\infty}{\text{Limit}} \; R_n \left(\frac{p}{q}\right)^{-\alpha n} = \underset{n\to\infty}{\text{Limit}} \frac{1 + U_n \frac{q}{p - q}}{1 + V_n \frac{p}{p - q}}$$

$$= 0 \quad \text{if } \alpha > \frac{\ln p}{\ln q}$$

$$= 1 \quad \text{if } \alpha < \frac{\ln p}{\ln q}$$

$$= \frac{1}{1 + \frac{p}{p - q}}$$

$$= \frac{p - q}{2p - q} \quad \text{if } \alpha = \frac{\ln p}{\ln q}$$

Then since $R_n^* \geq R_n(f)$, R_n^* increases asymptotically at least as fast as $\left(\frac{p}{q}\right)^{n \frac{\ln p}{\ln q}}$.

With the previous assumptions regarding the optimal form of the algorithm, we can now state that

$$\operatorname*{Limit}_{n\to\infty} \frac{\ln R_n^*}{n} \geq \frac{\ln p}{\ln q} \ln \frac{p}{q} \quad .$$

VI. DETERMINATION OF R_n^* AND f^* FOR SMALL n

To establish asymptotic bounds on R_n^*, it was necessary to assume that from some n on, optimal algorithms were such that $R_n^* > 1$ and $\rho(s) < s$ for $s > 1$.

To test this assumption, $R_n(f)$ was calculated for n = 1,2,3, and 4. For each case R_n^* was determined algebraically. For the values of $p \in (\frac{1}{2}, 1)$ it was found that

$$R_1^* = R_2^* = 1 \qquad \text{for any } f$$

$$R_3^* = \frac{p + q^3}{q + p^3} \qquad \text{with } f^* = \langle 1,1,2 \rangle$$

$$R_4^* = \frac{1 - pq - 2pq^2}{1 - pq - 2p^2q} \qquad \text{with } f^* = \langle 1,1,2,2 \rangle$$

The calculations used to determine these values of R^* are at Appendix A.

For values of n from 5 to 9, several values of $p > \frac{1}{2}$ were chosen and the search was performed using an IBM 360/67 in double precision. The program is at appendix B. For the vicinity of 1, values of p of .99 and .999 were used. In the vicinity of $\frac{1}{2}$, the optimal algorithm was determined by a Taylor series expansion around $\frac{1}{2} + \epsilon$, and neglecting terms of order ϵ^2. By neglecting terms of order ϵ^2,

$$R_n(f) = \frac{F_n(\frac{1}{2}) - \epsilon F_n'(\frac{1}{2})}{F_n(\frac{1}{2}) + \epsilon F_n'(\frac{1}{2})}$$

$$= \frac{1 - \epsilon \dfrac{F_n'(\frac{1}{2})}{F_n(\frac{1}{2})}}{1 + \epsilon \dfrac{F_n'(\frac{1}{2})}{F_n(\frac{1}{2})}}$$

To maximize $R_n(f)$, minimize $\dfrac{F_n'(\frac{1}{2})}{F_n(\frac{1}{2})}$.

Since F is a polynomial in x of degree less than or equal to n,

$$F_n(x) = \sum_{j=0}^{n} a(n,j)x^j,$$

where a(•,•) denotes the coefficients of the polynomial. Differentiating,

$$F_n'(x) = \sum_{j=1}^{n} ja(n,j)x^{j-1},$$

forming the ratio yields,

$$\frac{F_n'(\tfrac{1}{2})}{F_n(\tfrac{1}{2})} = \frac{\sum_{j=1}^{n} ja(n,j)(\tfrac{1}{2})^{j-1}}{\sum_{j=0}^{n} a(n,j)(\tfrac{1}{2})^{j}}$$

$$= \frac{\sum_{j=1}^{n} ja(n,j)2^{n-j+1}}{\sum_{j=0}^{n} a(n,j)2^{n-j}}$$

The program for the search of the ratio of these polynomials is at Appendix C.

The results are summarized in Table I and figures 7,8, and 9. They seem to confirm the assumptions made in proposition 2 and further indicate that the optimal form of the algorithm is probably of the form

$$f = \langle 1,1,\ldots,1,2,\ldots,2,3,\ldots,3,4,\ldots\rangle$$

with the lengths of the blocks of constant $\rho(s)$ depending on n and p.

Computer run time precludes a complete search of algorithms for n much larger than 9; however, it may be useful to assume that f* is of the form $\langle 1,..,1,2,\ldots,2,3,\ldots\rangle$ and to maximize R_n by manipulating the lengths of blocks of the ρ's. Such an algorithm is also intuitively appealing. Near the initial state the information content of an event is low as characterized by $\rho(s) = 1$. As the automaton approaches the decision

point (absorption), $\rho(s)$ increases as if the information content of a negative event had increased. In some respects this seems to be similar to the human decision process.

<table>
<tr><th>p</th><th>n=1</th><th>n=2</th><th>n=3</th><th>n=4</th><th>n=5</th><th>n=6</th><th>n=7</th><th>n=8</th><th>n=9</th></tr>
<tr><td>≈0.50</td><td rowspan="11">$\langle 1 \rangle$</td><td rowspan="11">$\langle 1,1 \rangle$ or $\langle 1,2 \rangle$</td><td rowspan="11">$\langle 1,1,2 \rangle$</td><td rowspan="11">$\langle 1,1,2,2 \rangle$</td><td rowspan="11">$\langle 1,1,2,2,3 \rangle$</td><td rowspan="3">$\langle 1,1,1,2,2,3 \rangle$</td><td rowspan="4">$\langle 1,1,1,2,2,3,3 \rangle$</td><td rowspan="2">$\langle 1,1,1,1,2,2,3,3 \rangle$</td><td>$\langle 1,1,1,1,2,2,2,2,3 \rangle$</td></tr>
<tr><td>0.55</td><td rowspan="2">$\langle 1,1,1,1,2,2,3,3,3 \rangle$</td></tr>
<tr><td>0.60</td><td>$\langle 1,1,1,2,2,2,2,3 \rangle$</td></tr>
<tr><td>0.65</td><td rowspan="8">$\langle 1,1,2,2,3,3 \rangle$</td><td rowspan="2">$\langle 1,1,1,2,2,3,3,3 \rangle$</td><td>$\langle 1,1,1,2,2,2,2,3,3 \rangle$</td></tr>
<tr><td>0.70</td><td>$\langle 1,1,2,2,2,2,3 \rangle$</td><td>$\langle 1,1,1,2,2,3,3,3,3 \rangle$</td></tr>
<tr><td>0.75</td><td rowspan="6">$\langle 1,1,2,2,3,3,3 \rangle$</td><td>$\langle 1,1,2,2,2,2,3,3 \rangle$</td><td>$\langle 1,1,2,2,2,2,3,3,3 \rangle$</td></tr>
<tr><td>0.80</td><td>$\langle 1,1,2,2,3,3,3,3 \rangle$</td><td>$\langle 1,1,2,2,3,3,3,3,3 \rangle$</td></tr>
<tr><td>0.85</td><td rowspan="4">$\langle 1,1,2,2,3,3,3,4 \rangle$</td><td rowspan="4">$\langle 1,1,2,2,3,3,3,4,4 \rangle$</td></tr>
<tr><td>0.90</td></tr>
<tr><td>0.95</td></tr>
<tr><td>≈1.00</td></tr>
</table>

Table 1.

Optimal Algorithm $f^* = \langle \rho(1),\ldots,\rho(n) \rangle$

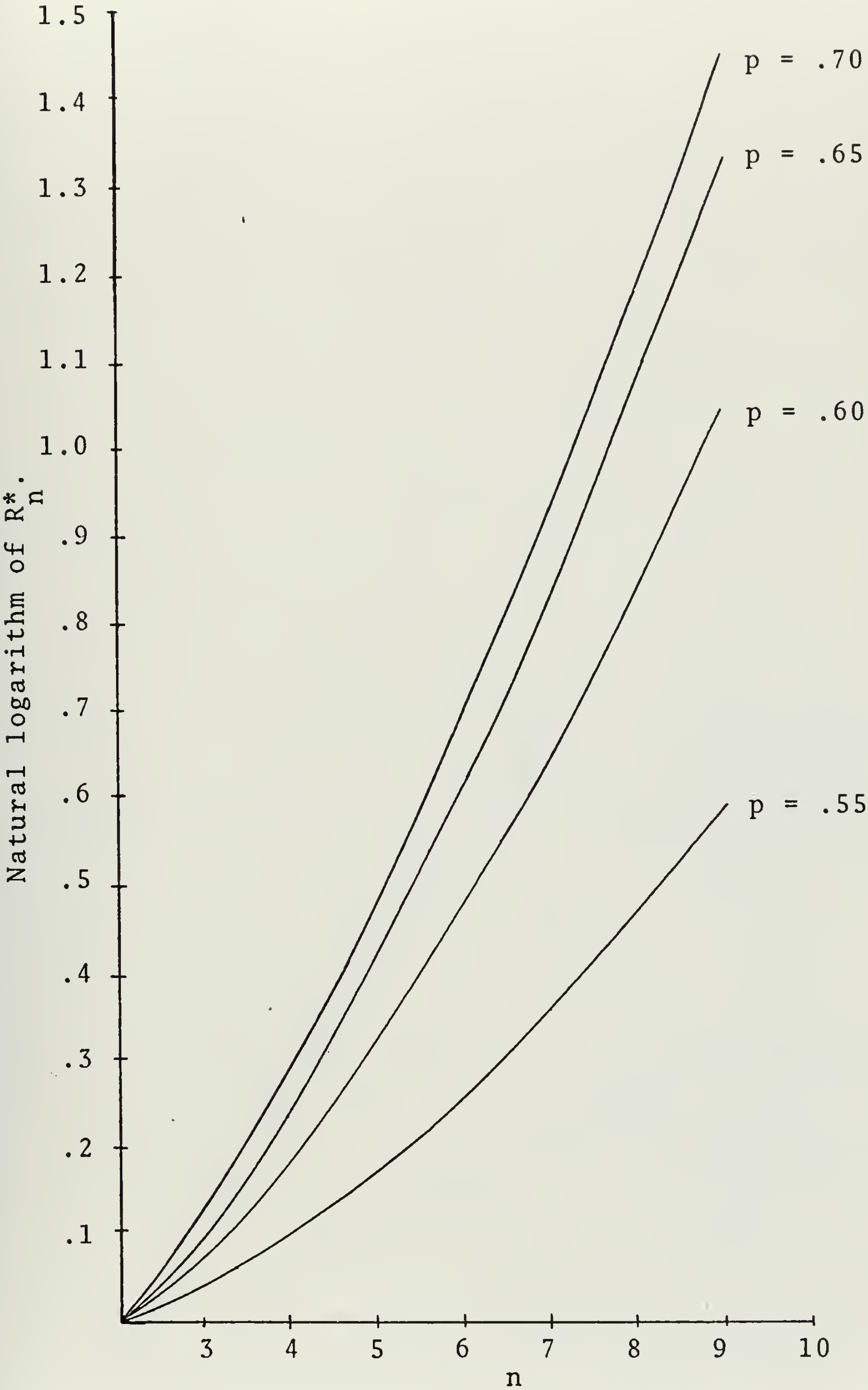


Figure 7.
Natural logarithm of R_n^* versus n.

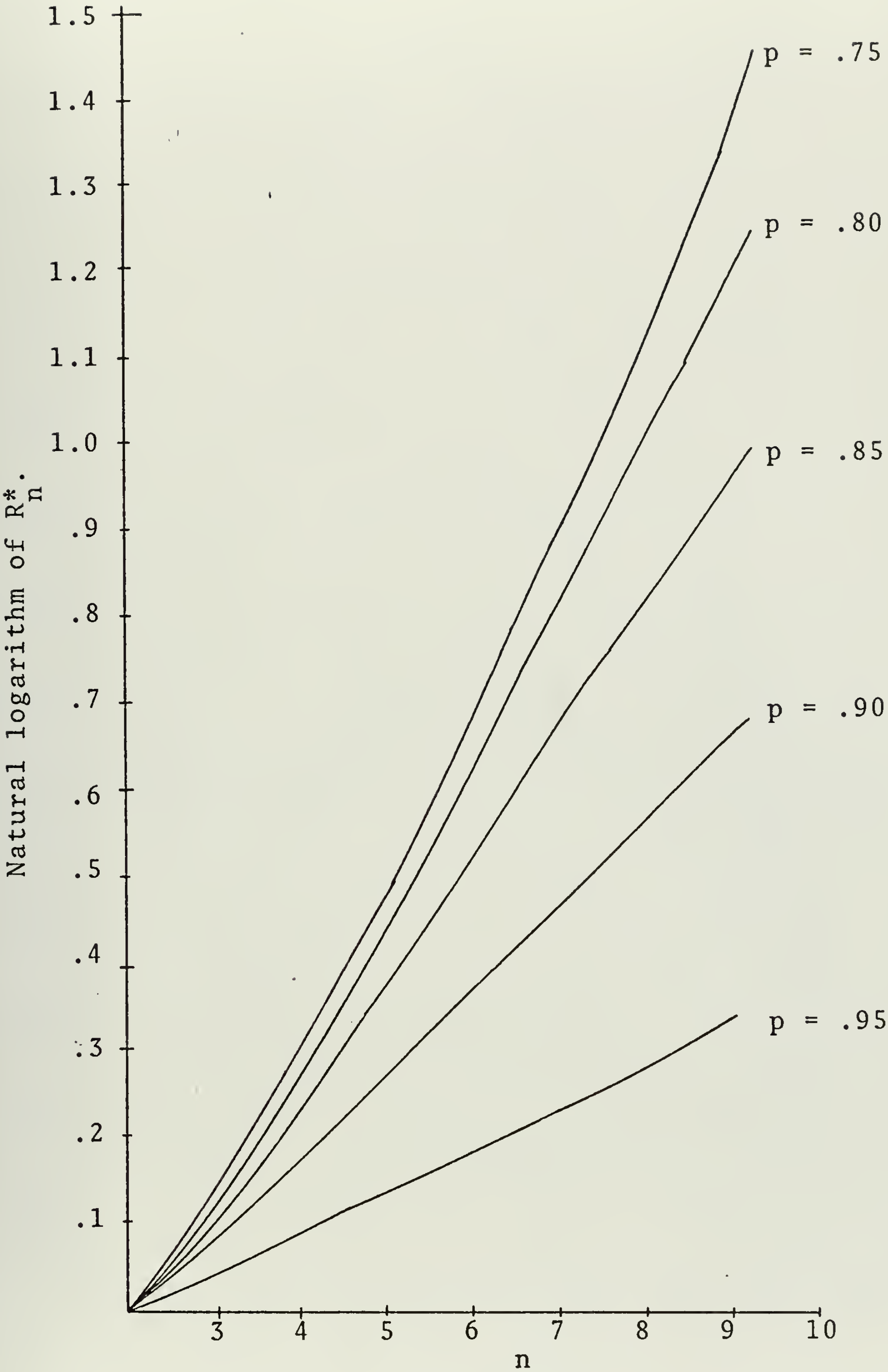


Figure 8.
Natural logarithm of R_n^* versus n.

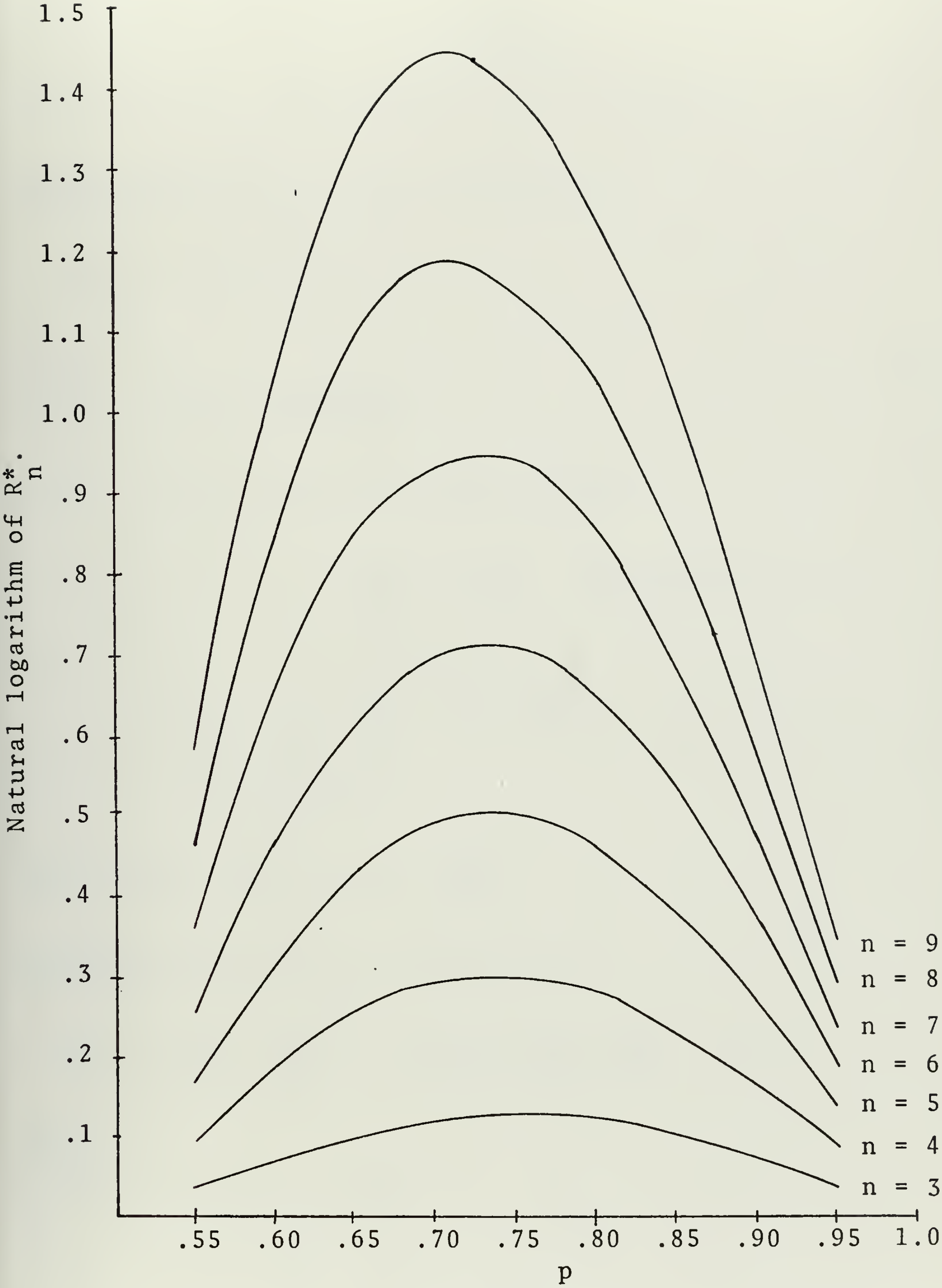


Figure 9.
Natural logarithm of R_n^* versus p.

APPENDIX A

Calculation of R* for n=3 and n=4.

For the case when n = 3, $f_3{*} = \langle 1,1,2 \rangle$, and

$$R_3{*} = \frac{1 - 2p + 3p^2 - p^3}{1 - p + p^3}$$

Proof:

If $f_3^{(1)} = \langle 1,1,1 \rangle$, then $R_3^{(1)} = 1$.

If $f_3^{(2)} = \langle 1,1,2 \rangle$, then $R_3^{(2)} = \dfrac{1 - 2p + 3p^2 - p^3}{1 - p + p^3}$

$$3p > 1 + 2p^2 \qquad \text{for all } \tfrac{1}{2} < p < 1.$$

$$3p^2 > p + 2p^3$$

$$1 - 2p + 3p^2 - p^3 > 1 - p + p^3$$

$$\frac{1 - 2p + 3p^2 - p^3}{1 - p + p^3} > 1$$

Therefore $R_3^{(2)} > R_3^{(1)}$, and $R_3{*} = R_3^{(2)}$.

For the case of n = 4, $f\,{*} = \langle 1,1,2,2 \rangle$, and

$$R_4{*} = \frac{1 - pq - 2pq^2}{1 - pq - 2p^2q} = \frac{1 - 3p + 5p^2 - 2p^3}{1 - p - p^2 + 2p^3}$$

Proof:

If $f^{(1)} = \langle 1,1,1,1 \rangle$, then $R_4^{(1)} = 1$.

$$6p - 4p^2 > 2 \qquad \text{for all } \tfrac{1}{2} < p < 1$$

$$6p^2 - 4p^3 > 2p; \quad \text{adding } 1 - 3p - p^2 + 3p^3$$

$$1 - 3p + 5p^2 - 2p^3 > 1 - p - p^2 + 2p^3$$

$$\frac{1 - 3p + 5p^2 - 2p^3}{1 - p - p^2 + 2p^3} > 1$$

Thus $R_4^* > R_4^{(1)}$

If $f^{(2)} = \langle 1,1,1,2\rangle$, then $R_4^{(2)} = \dfrac{1 - 3p + 4p^2 - p^3}{1 - 2p + p^2 + p^3}$

$$6p + 15p^3 > 1 + 14p^2 + 6p^4 \qquad \text{for all } \tfrac{1}{2} < p < 1$$

$$6p^2 + 15p^4 > p + 14p^3 + 6p^5$$

adding $1 - 5p + 6p^2 - 14p^3 - 9p^4 + 9p^5 - 2p^6$

$$1 - 5p + 12p^2 - 14p^3 + 6p^4 + 3p^5 - 2p^6$$
$$> 1 - 4p + 6p^2 - 9p^4 + 9p^5 - 2p^6$$

$$(1 - 3p + 5p^2 - 2p^3)(1 - 2p + p^2 + p^3)$$
$$> (1 - 3p + 4p^2 - p^3)(1 - p - p^2 + 2p^3)$$

$$\frac{1 - 3p + 5p^2 - 2p^3}{1 - p - p^2 + 2p^3} > \frac{1 - 3p + 4p^2 - p^3}{1 - 2p + p^2 + p^3}$$

Thus $R_4^* > R_4^{(2)}$

If $f^{(3)} = \langle 1,1,1,3\rangle$, $R_4^{(3)} = \dfrac{1 - 3p + 5p^2 - 3p^3 + p^4}{1 - 2p + 2p^2 - p^3 + p^4}$

$$6p+25p^3+14p^5 > 1+16p^2+24p^4+4p^6 \qquad \text{for all } \tfrac{1}{2} < p < 1.$$

$$6p^2+25p^4+14p^6 > p+16p^3+24p^5+4p^7$$

Adding $1-5p+7p^2-19p^3-7p^4-12p^5-7p^6-2p^7$

$$1-5p+13p^2-19p^3+18p^4-12p^5+7p^6-2p^7 > 1-4p+7p^2-3p^3-7p^4+12p^5-7p^6+2p^7$$

$$(1-3p+5p^2-2p^3)(1-2p+2p^2-p^3+p^4) > (1-3p+5p^2-3p^3+p^4)(1-p-p^2+2p^3)$$

$$\frac{1-3p+5p^2-2p^3}{1-p-p^2+2p^3} > \frac{1-3p+5p^2-3p^3+p^4}{1-2p+2p^2-p^3+p^4}$$

Thus $R_4^* > R_4^{(3)}$.

If $f^{(4)} = \langle 1,1,2,1,\rangle$, $R_4^{(4)} = \dfrac{1-3p+5p^2-3p^3+p^4}{1-2p+2p^2-p^3+p^4}$

$$= R_4^{(3)}$$

Thus, $R_4{*} > R_4^{(4)}$.

If $f^{(5)} = \langle 1,1,2,3\rangle$, $R_4^{(5)} = \dfrac{1-3p+6p^2-4p^3+p^4}{1-p+p^4}$

$$10p+14p^3 > 2+18p^2+4p^4$$

$$10p^4+14p^6 > 2p^3+18p^5+4p^7$$

Adding $1-4p+8p^2-7p^3-7p^4-3p^5-9p^6-2p^7$

$$1-4p+8p^2-7p^3+3p^4-3p^5+5p^6-2p^7 > 1-4p+8p^2-5p^3-7p^4+15p^5-9p^6+2p^7$$

$$(1-3p+5p^2-2p^3)(1-p+p^4) > (1-3p+6p^2-4p^3+p^4)(1-p-p^2+2p^3)$$

$$\frac{1-2p+5p^2-2p^3}{1-p-p^2+2p^3} > \frac{1-3p+6p^2-4p^3+p^4}{1-p+p^4}$$

Thus $R_4{*} > R_4^{(5)}$ which completes the proof.

APPENDIX B

```
//AND12514 JOB (2514,1242FT,RL22),'ANDERSON',TIME=(4,0)
// EXEC FORTCLG,REGION.GO=100K
//FORT.SYSIN DD *
      IMPLICIT REAL*8(A,B,C,D,R,S)
      DIMENSION A(11),B(11)
      A(1)=1.
      B(1)=1.
      A(2)=1.
      B(2)=1.
   30 READ(5,2000,END=20)C
      WRITE(6,3005) C
      D=1.-C
      A(3)=1.-C*D
      B(3)=1.-C*D
      SMAX=0.
      DO 10 I3=1,2
      DO 10 I4=1,3
      DO 10 I5=1,4
      DO 10 I6=1,5
      DO 10 I7=1,6
      DO 10 I8=1,7
      DO 10 I9=1,8
      A(4)=A(3)-C**I3*D*A(3-I3)
      B(4)=B(3)-D**I3*C*B(3-I3)
      A(5)=A(4)-C**I4*D*A(4-I4)
      B(5)=B(4)-D**I4*C*B(4-I4)
      A(6)=A(5)-C**I5*D*A(5-I5)
      B(6)=B(5)-D**I5*C*B(5-I5)
      A(7)=A(6)-C**I6*D*A(6-I6)
      B(7)=B(6)-D**I6*C*B(6-I6)
      A(8)=A(7)-C**I7*D*A(7-I7)
      B(8)=B(7)-D**I7*C*B(7-I7)
      A(9)=A(8)-C**I8*D*A(8-I8)
      B(9)=B(8)-D**I8*C*B(8-I8)
      A(10)=A(9)-C**I9*D*A(9-I9)
      B(10)=B(9)-D**I9*C*B(9-I9)
      R=B(10)/A(10)
      IF(R.LT.SMAX) GO TO 10
      SMAX=R
      K3=I3
      K4=I4
      K5=I5
      K6=I6
      K7=I7
      K8=I8
      K9=I9
   10 CONTINUE
      WRITE (6,3000) SMAX
      WRITE(6,3001) K3,K4,K5,K6,K7,K8,K9
 3001 FORMAT(T10,7I6,//)
      GO TO 30
 2000 FORMAT(F6.4)
 3000 FORMAT (T10,F15.7)
 3005 FORMAT(T10,'F= ',F6.4)
   20 STOP
      END
```

```
      IMPLICIT INTEGER*4(R,S)
      DIMENSION R(11,11),S(11,11)
      DATA R/121*0/,S/121*0/
      R(2,2)=1
      R(3,2)=1
      S(2,2)=1
      S(3,2)=1
      S(2,3)=-1
      S(3,3)=-1
      A2=99999.
      I2=1
      DO 20 K=1,I2
   20 R(4,K)=R(3,K)
      M=I2+1
      DO 21 K=M,11
   21 R(4,K)=R(3,K)-S(3-I2,K-I2)
      DO 22 K=2,11
   22 S(4,K)=R(4,K)-R(4,K-1)
      DO 400 I3=1,2
      DO 30 K=1,I3
   30 R(5,K)=R(4,K)
      M=I3+1
      DO 31 K=M,11
   31 R(5,K)=R(4,K)-S(4-I3,K-I3)
      DO 32 K=2,11
   32 S(5,K)=R(5,K)-R(5,K-1)
      DO 400 I4=1,3
      DO 40 K=1,I4
   40 R(6,K)=R(5,K)
      M=I4+1
      DO 41 K=M,11
   41 R(6,K)=R(5,K)-S(5-I4,K-I4)
      DO 42 K=2,11
   42 S(6,K)=R(6,K)-R(6,K-1)
      DO 400 I5=1,4
   50 R(7,K)=R(6,K)
      M=I5+1
      DO 51 K=M,11
   51 R(7,K)=R(6,K)-S(6-I5,K-I5)
      DO 52 K=2,11
   52 S(7,K)=R(7,K)-R(7,K-1)
      DO 400 I6=1,5
      DO 60 K=1,I6
   60 R(8,K)=R(7,K)
      M=I6+1
      DO 61 K=M,11
   61 R(8,K)=R(7,K)-S(7-I6,K-I6)
      DO 62 K=2,11
   62 S(8,K)=R(8,K)-R(8,K-1)
      DO 400 I7=1,6
      DO 70 K=1,I7
   70 R(9,K)=R(8,K)
      M=I7+1
      DO 71 K=M,11
   71 R(9,K)=R(8,K)-S(8-I7,K-I7)
      DO 72 K=2,11
   72 S(9,K)=R(9,K)-R(9,K-1)
      DO 400 I8=1,7
      DO 80 K=1,I8
   80 R(10,K)=R(9,K)
      M=I8+1
      DO 81 K=M,11
   81 R(10,K)=R(9,K)-S(9-I8,K-I8)
      DO 82 K=2,11
   82 S(10,K)=R(10,K)-R(10,K-1)
      DO 400 I9=1,8
      DO 90 K=1,I9
   90 R(11,K)=R(10,K)
      M=I9+1
      DO 91 K=M,11
   91 R(11,K)=R(10,K)-S(10-I9,K-I9)
      DO 92 K=2,11
```

```
   92 S(11,K)=R(11,K)-R(11,K-1)
      R1=0
      R2=0
      N7=9
      DO 27  J=1,N7
      J1=J
      J2=N7-J1+1
   27 R1=R1+J1*2**J2*R(11,J1+2)
      N7=N7+1
      DO 28  J=1,N7
      J1=J
      J2=N7-J1
   28 R2=R2+2**J2*R(11,J1+1)
      A1=R1
      A1=A1/R2
      IF(A1.GT.A2) GO TO 400
      A2=A1
      L2=I2
      L3=I3
      L4=I4
      L5=I5
      L6=I6
      L7=I7
      L8=I8
      L9=I9
      R12=R(11,2)
      R13=R(11,3)
      R14=R(11,4)
      R15=R(11,5)
      R16=R(11,6)
      R17=R(11,7)
      R18=R(11,8)
      R19=R(11,9)
      R110=R(11,10)
      R111=R(11,11)
  400 CONTINUE
      WRITE(6,2001) R12,R13,R14,R15,R16,R17,R18,R19,R110
     1,R111
      WRITE(6,2002) L2,L3,L4,L5,L6,L7,L8,L9,A2
 2001 FORMAT(T10,'COEFFICIENTS',10I6)
 2002 FORMAT(T10,'INDEX VALUES',8I5,/,T10,'MIN=',F14.8,//)
      STOP
      END
```

REFERENCES

[1] Cover, Thomas M., "Hypothesis Testing with Finite Statistics," The Annals of Mathematical Statistics, Vol. 40, No. 3, p. 828-835.

[2] Robbins, Herbert, "A Sequential Decision Problem With a Finite Memory," Proc. Nat. Acad. Sci., U.S.A., Vol. 42, p. 920-923.

[3] Hellman, Martin E. and Cover, Thomas M., "Learning With Finite Memory," The Annals of Mathematical Statistics, Vol. 41, No. 3, p. 765-782.

[4] Hellman, Martin E., "The Effects of Randomization on Finite-Memory Decision Schemes," IEEE Transactions on Information Theory, IT-18, No. 4, p. 499-502.

[5] Bhat, U. Narayan, Elements of Applied Stochastic Processes, Wiley, 1972.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Chief of Personnel Operations ATTN: OPXC Department of the Army Washington, D.C. 20315	1
3.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
4.	Professor Bruno Shubert (thesis advisor) Operations Analysis Department Naval Postgraduate School Monterey, California 93940	1
5.	MAJ Calvin M. Anderson (student) 2821 Ransford Avenue Pacific Grove, California 93940	1

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

INATING ACTIVITY (Corporate author) aval Postgraduate School onterey, California 93940	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

ORT TITLE

'esting a Simple Symmetric Hypothesis by a Finite-Memory Deterministic Algorithm

:RIPTIVE NOTES (Type of report and, inclusive dates)

aster's Thesis; September 1973

HOR(S) (First name, middle initial, last name)

alvin Marion Anderson

ORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
;eptember 1973		5

NTRACT OR GRANT NO. OJECT NO.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

TRIBUTION STATEMENT

Approved for public release; distribution unlimited.

PPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Naval Postgraduate School Monterey, California 93940

STRACT

A class of finite-memory deterministic algorithms is introduced and investigated. Optimum algorithms are found for a small number of states (up to 21) and an asymptotic bound on error probability if obtained for a large number of states. The algorithms provide their own stopping rule.

UNCLASSIFIED
Security Classification

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Automaton						
Finite-Memory Deterministic Algorithm						
Decision-making						
Hypothesis Testing with Finite Memory						

DD FORM 1 NOV 65 1473 (BACK)
0101-807-6821

UNCLASSIFIED
Security Classification

A-31409

23 NOV 73 21853

Thesis 143478
A4453 Anderson
c.1 Testing a simple symmetric hypothesis by a finite-memory deterministic algorithm.
23 NOV 73 21853

Thesis 143478
A4453 Anderson
c.1 Testing a simple symmetric hypothesis by a finite-memory deterministic algorithm

thesA4453
Testing a simple symmetric hypothesis by
3 2768 000 99127 7
DUDLEY KNOX LIBRARY